**Day 9: Object-Oriented Programming in Java - Classes and Objects!**

Over the past few days, we've covered the basics of Java, including variables, data types, operators, control structures, and functions. Today, we're going to dive into one of the most powerful concepts in Java: object-oriented programming (OOP)!

**What is OOP?**

Object-oriented programming is a programming paradigm that revolves around the concept of objects and classes. In OOP, you define classes that represent real-world objects or abstract concepts, and then create objects that are instances of those classes.

**Classes and Objects**

In Java, a class is a blueprint or a template that defines the properties and behavior of an object. An object, on the other hand, is an instance of a class, and it has its own set of attributes (data) and methods (functions).

Here's an example:

public class Dog {

private String name;

private int age;

public Dog(String name, int age) {

this.name = name;

this.age = age;

}

public void bark() {

System.out.println("Woof!");

} }

public class Main {

public static void main(String[] args) {

Dog myDog = new Dog("Fido", 3);

myDog.bark(); // Output: Woof!

} }

**Challenges**

Create a class called Rectangle with attributes width and height, and methods area() and perimeter(). Create an object of the Dog class and call its bark() method.

**Code**

You can find the code for today's challenges in the Day9 folder of this repository.

**Join the Conversation**

How was your experience with classes and objects in Java today? Did you encounter any challenges or have any questions? Share your thoughts in the comments below!